

Homework 2 - Cellular Automata

Claudia Lizeth Hernández Ramírez

8 de septiembre de 2021

Resumen

No se encuentra una tendencia específica de comportamiento. Para el rango de 0.2 a 0.5 parece como un comportamiento específico, sin embargo como era de esperarse a valores de 0.1 y 0.9 las probabilidades infinitas son muy diferentes.

1. Introducción

En este trabajo se pretende establecer la posibilidad de crear vida infinita en función de la posibilidad inicial de crear vida.

2. Desarrollo

Teniendo el código base [2] comencé eliminando lo que no era necesario para la tarea. Posteriormente como se mencionó en clase necesitaba implementar dos ciclos `for` uno para variar la `p` que es la posibilidad en la que mi código generará vida por primera ocasión, en mi caso decidí hacerlo partiendo de 0.1 hasta llegar a 0.9; el segundo `for` era necesario para las réplicas, en mi caso fueron 30 valores.

Para comenzar a determinar si quedaba alguien con vida al finalizar el código, con la ayuda de un `if` establecí lo siguiente: si mi variable `vivos` resultaba en un valor mayor a 0, `vivos` tomaría el valor de 1 y lo imprimiría como un 1, de lo contrario lo imprimiría como un 0.

Hasta ese momento todo parecía ir bastante bien, el reto fue lograr que esos 1 y 0 se fueran contabilizando para lograr sacar la estadística y probabilidad de creación de vida infinita. Después de dos días tratando de obtener información y de varios intentos fallidos, probé con algo que me dio resultado. "`datos`" representaba un data frame vacío. `unos` y `ceros` eran una variable de tipo numérico sin un valor definido. Al terminar de ejecutarse mi `if`, llamé a mi variable `unos` y lo definí como un vector el cual se conformaría de la suma de los valores de "`vivos`" que fueran mayores a 0, `ceros` por otro lado haría lo mismo, solo que con "`vivos`" igual a 0. Después imprimiría mi vector `unos`. Una nueva variable llamada "`dfunos`" por "`data frame unos`" entró al juego, ya que ahí es donde se guardan mis valores de 1 y 0.

Un nuevo problema era que no logré acomodar esos valores de una manera en la que pudiera trabajar, sabía cómo estaban constituidos, sin embargo, no logré acomodarlos de una forma óptima, por lo que se me ocurrió exportar a excel y acomodarlos de una mejor forma para posteriormente exportar esa misma información a Rstudio[3]. Con el paquete '`ggplot`' generé una gráfica de puntos '`geompoint`' que posteriormente uní con un '`geomline`'.

probabilidad inicial	cantidad de unos	probabilidad infinito
0.1	0.2667	8
0.2	0.1333	4
0.3	0.1667	5
0.4	0.2000	6
0.5	0.1667	5
0.6	0.1667	5
0.7	0.2000	6
0.8	0.2333	7
0.9	0.400	12

Cuadro 1: Datos que se Obtienen en Rstudio

Finalmente este es el código con el que generé toda la tarea[1]:

```
library(parallel)
library(openxlsx)
dim = 20
num = dim^2
dur = 10
p = c(0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90)
datos = data.frame()
unos = numeric()
ceros = numeric()
for (inicial in p) {
  for (replica in 1:30) { # minimo 30 1:30
    actual = matrix(round(runif(num) < p), nrow=dim, ncol=dim, byrow=TRUE)
    paso = function(pos) {
      fila = floor((pos - 1) / dim) + 1
      columna = ((pos - 1) %% dim) + 1
      vecindad = actual[max(fila - 1, 1) : min(fila + 1, dim),
                        max(columna - 1, 1) : min(columna + 1, dim)]
      return(1 * ((sum(vecindad) - actual[fila, columna]) == 3))
    }
    cluster <- makeCluster(detectCores() - 1)
    clusterExport(cluster, "dim")
    clusterExport(cluster, "paso")

    for (iteracion in 1:dur) {
      clusterExport(cluster, "actual")
      siguiente <- parSapply(cluster, 1:num, paso)
      vivos = sum(siguiente)
      cat(inicial, replica, iteracion, vivos, '\n') #concatenar
      actual <- matrix(siguiente, nrow=dim, ncol=dim, byrow=TRUE)
    }
    if (vivos > 0) {
      vivos = 1
      print(vivos)
    } else {
      vivos = 0
      print(0)
    }
  }
  unos = c(unos, sum(vivos > 0)) #vector de 1s y 0s
  ceros = c(ceros, sum(vivos == 0))
  stopCluster(cluster)
}
```

```

print(unos)
dfunos = data.frame()
dfunos = rbind(dfunos, unos)
}
write.xlsx(dfunos, file = "miarchivodf.xlsx")

library(ggplot2)#Grafica
library(openxlsx)
library(tidyverse)
library(dplyr)

ggplot(miarchivodf, aes(x = probini,
                        y = probinfini)) +
  geom_line(color="blue") +
  geom_point(shape=21, color="blue", fill="blue", size=3) +
  theme_gray() + labs (x = "Probabilidad inicial de poblaci n",
                      y = "Probabilidad de crear vida infinita",
                      title = "Juego de la vida")

```

Después de ejecutar el código anterior, se obtuvo la siguiente gráfica.

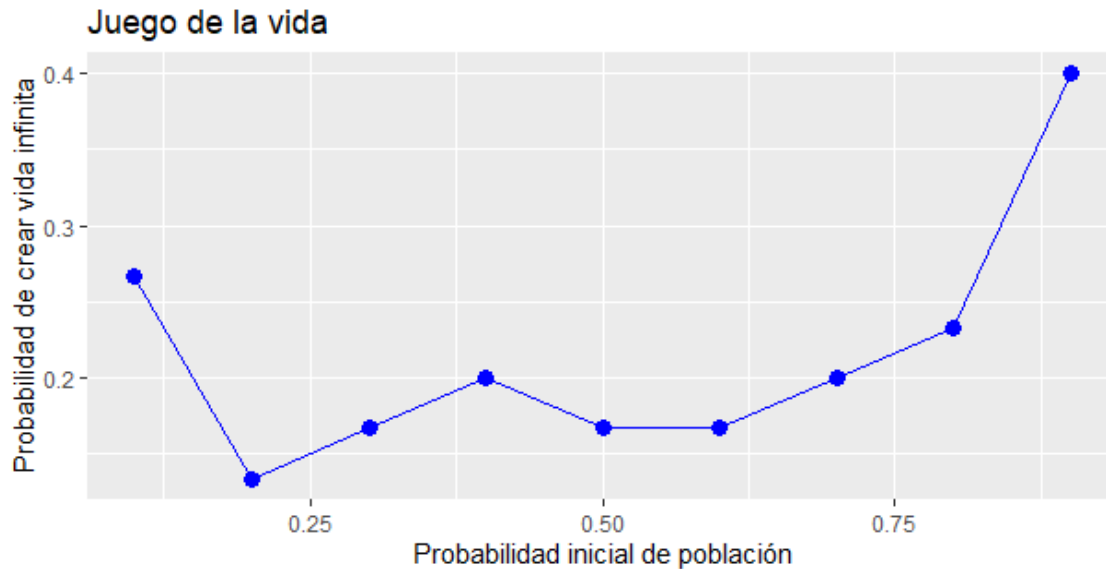


Figura 1: Gráfica de puntos. Probabilidad de crear vida infinita.

3. Conclusión

Con base en los resultados obtenidos en nuestra gráfica, podemos deducir que si comenzamos con una probabilidad inicial de población la probabilidad de generar vida inifnita aumenta en comparación del rango de 0.2 a 0.7, ya que sucede lo mismo en una probabilidad inicial de 0.9, en donde es casi un 50 % de probabilidad de generar vida infinita. El comportamiento que tiene la gráfica no muestra tendencia general, sino que tiene altos y bajos.

Referencias

- [1] Manual listings. URL https://es.wikibooks.org/wiki/Manual_de_LaTeX/Listados_de_c%C3%B3digo/Listados_con_listings.
- [2] Repositorio elisa schaeffer. URL <https://github.com/satuelisa/Simulation/blob/master/CellularAutomata/gameOfLife.R>.
- [3] Tablas overleaf. URL <https://www.youtube.com/watch?v=9cXHpub1Ac0>.